RUHR-UNIVERSITÄT BOCHUM

RUB

Protecting Cryptographic Instances on Reconfigurable Systems

XSWG 2014 – Xilinx Longmont Summit Retreat Center

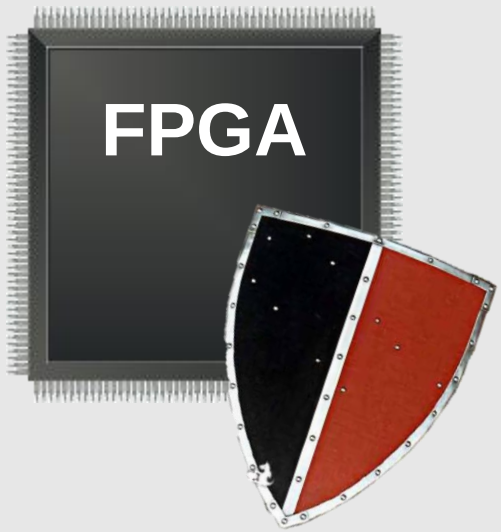
Tim Güneysu

Hardware Security Group
Horst Görtz Institute for IT-Security

28.10.2014

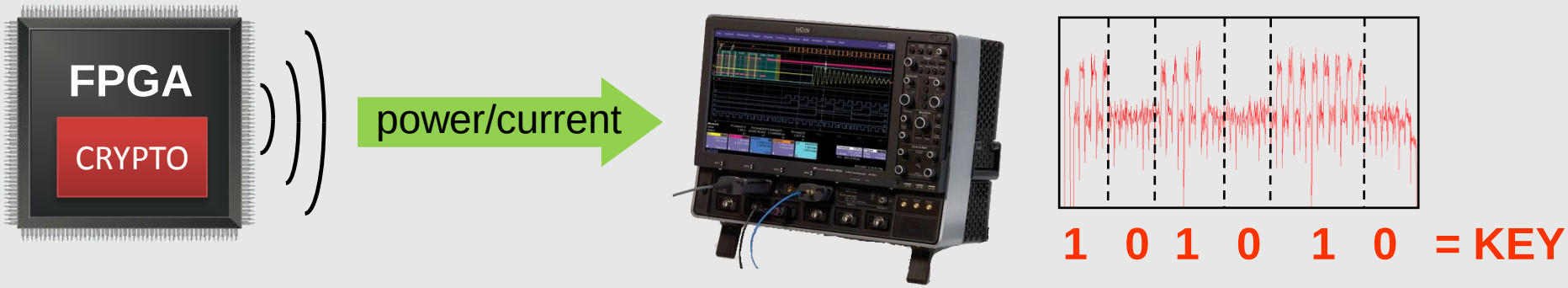
Agenda

- **Introduction and Motivation**
- Countermeasures for Reconfigurable Hardware
  - Noise Generation
  - Disaligning Clocks
  - Power Equalization
  - Memory Masking
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- Conclusions



Introduction and Motivation

- **Every cryptographic implementations needs countermeasures (CM) against physical attacks (here: power consumption as side-channel)**

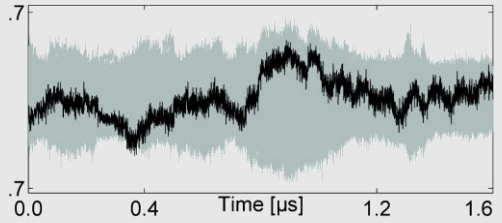


- **Designing and deploying a CM on a dedicated platform is costly**
  - Extra development + execution, additional ressources
  - For strong protection, **several CMs need to be combined**

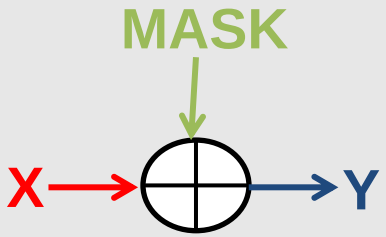
- **In an ideal world: Use prebuilt solutions from a set of generic CMs to establish (basic) SCA protection on your reconfigurable platform**

Introduction and Motivation: Power Analysis

- **Countermeasure: Hiding**
  - Manipulate amplitude of exploitable information
    - Decrease SNR using noise generator extensions
    - Decrease SNR using alternative logic styles
  - Runtime randomization (shuffling operations, irregular clocks)



- **Countermeasure: Masking**
  - Additive Masking (Boolean, multiplicative)
  - Data decomposition (secret sharing)
  - Blinding of secret parameters (exponent, scalars)
  - HW-based masking (random precharge, bus masking)



Protecting Cryptographic Instances in FPGAs

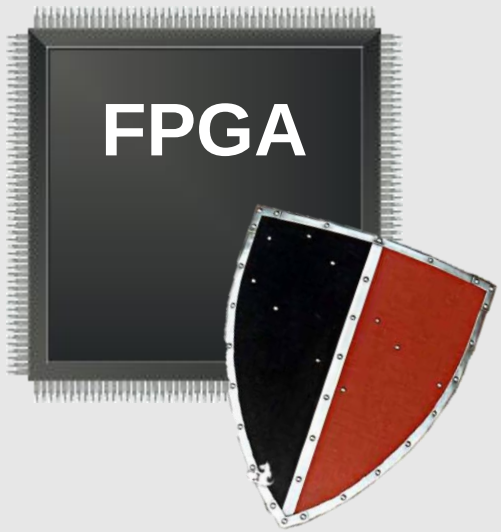
- **This talk: specifically tailored countermeasures for FPGAs**
  - Generically usable with most (symmetric) cryptosystems
  - Applicable to many (Xilinx) FPGA devices
  - Some can be even distributed as hard macros

- **Portfolio of countermeasures:**
  - FPGA-specific noise generators (using registers, memories, short circuits)
  - Clock disalignment using Digital Clock Managers (DCM)
  - Power Equalization using logic duplication
  - Memory masking in dual-ported memories

- **Integrate protected cryptographic cores into full system security concept**

Agenda

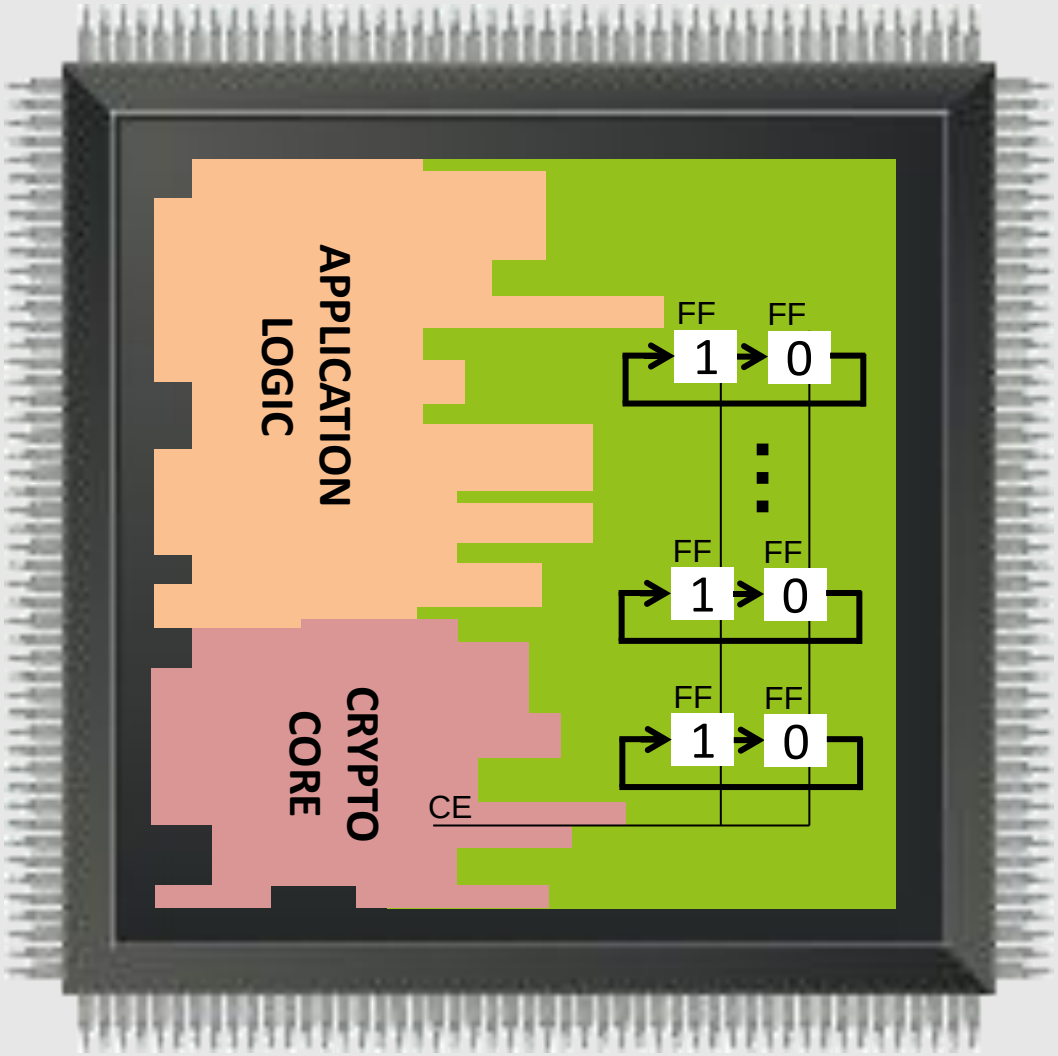
- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - **Noise Generation**
  - Disaligning Clocks
  - Power Equalization
  - Memory Masking
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- Conclusions



Implementing Noise Generators in FPGAs

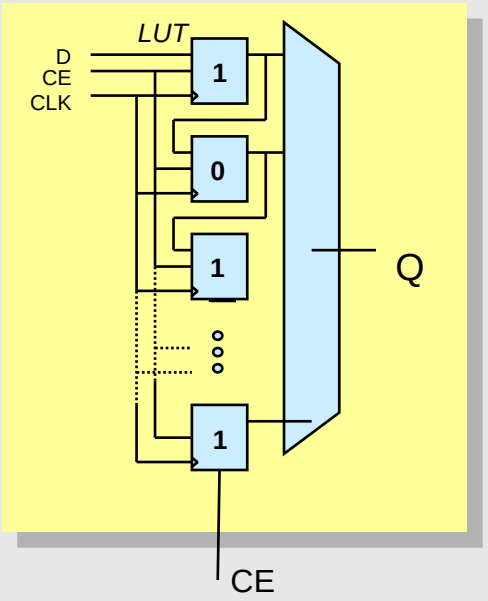
- **Common design**: application including cryptographic core

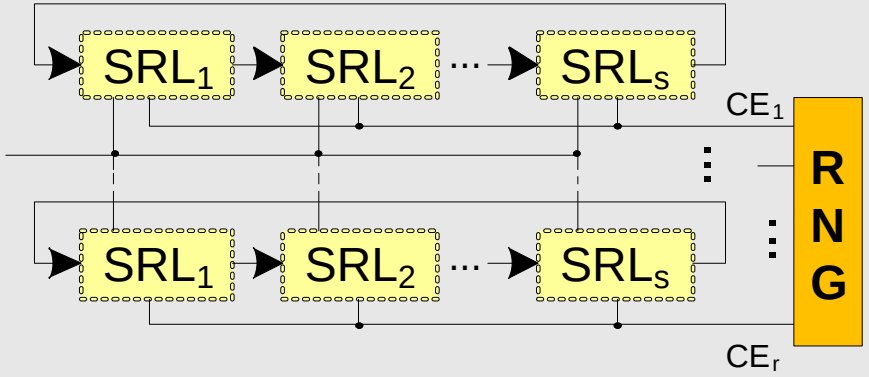
- **Noise generation strategy**
  - Configure remaining, routable slices (flip-flops) as cyclic shift registers
  - Preload sequence „01“ into shift registers
  - Run noise generator in synch with crypto core



Proposal #1: Using Shift Register LUTs (SRL)

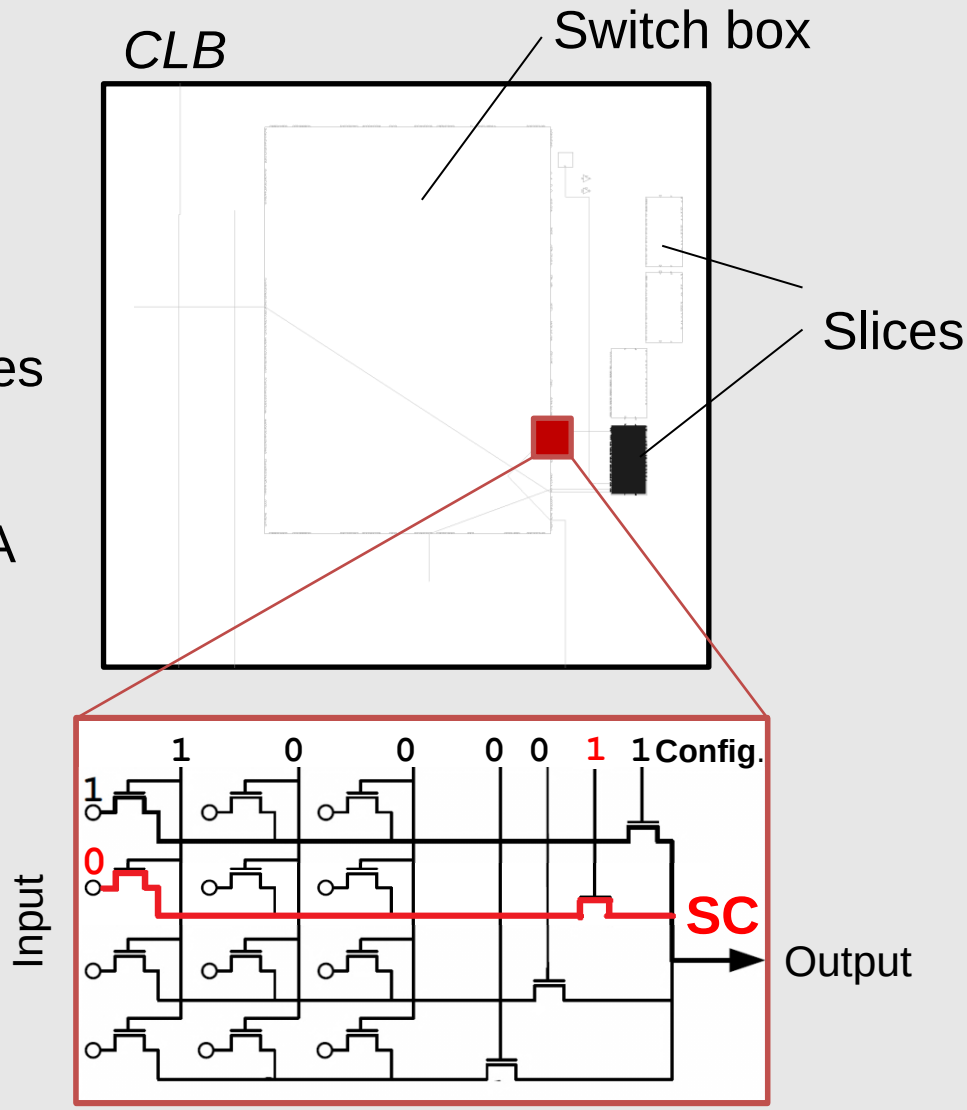
- Logic elements consist of LUTs and FFs
- Special (alternative) LUT function:
  Shift Register LUT (SRL)
  - *n*-bit register length (*n=16* or *n=64*)
  - Preload SRL with „01“ combination
- Create *r* cyclic rings using *s* cascaded SRLs
- SRLs are clocked according to free-running RNG





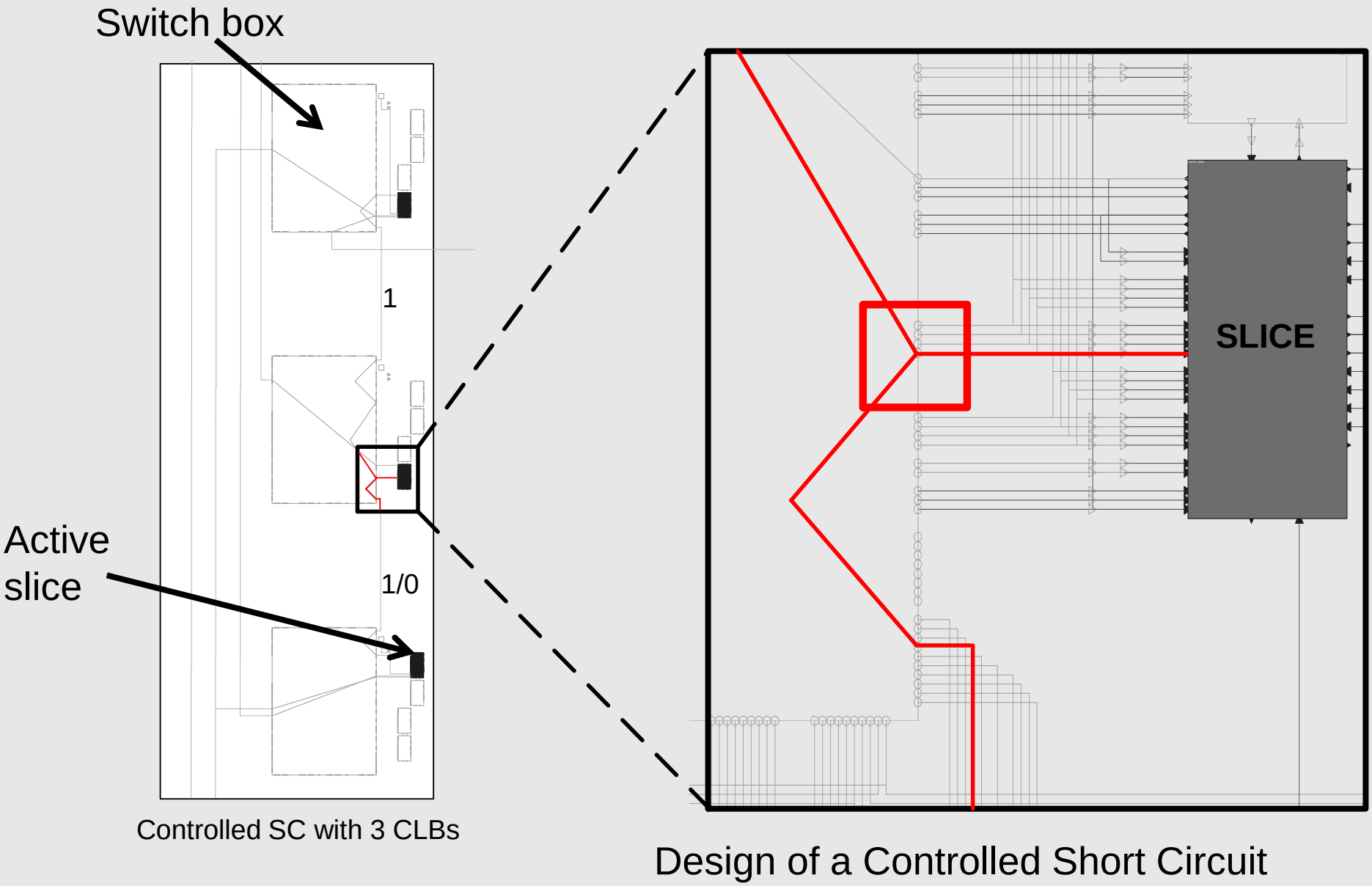
Proposal #2: Short Circuits in FPGAs

- Short circuits (SC) can be created in the FPGA‘s routing network [BKT10]
- SCs in output multiplexers of switch boxes
- Power restriction limits currents < 100 µA
- Establishing controlled SCs requires manual routing (via XDL)



Output multiplexer

Proposal #2: Short Circuits in FPGAs

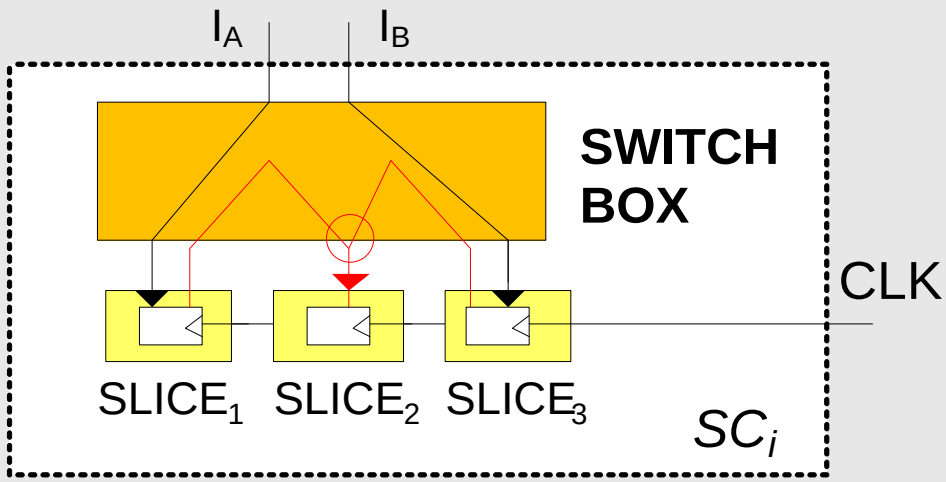


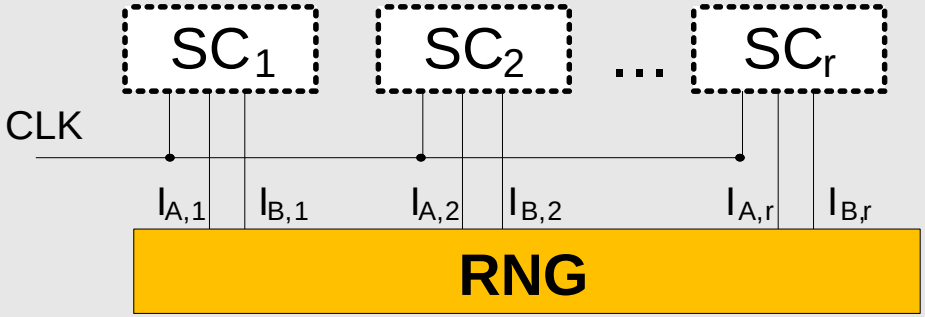
Controlled SC with 3 CLBs

Design of a Controlled Short Circuit

Proposal #2: Short Circuits in FPGAs

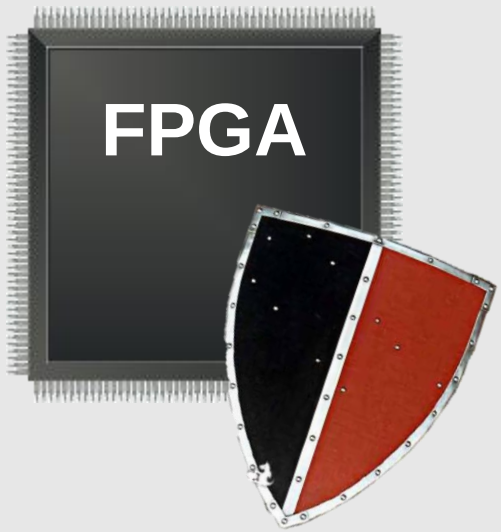
- Package controlled SC into hard macro
- Instantiate *r* controlled SC units on FPGA
- Distribute SCs among different power domains to distribute load





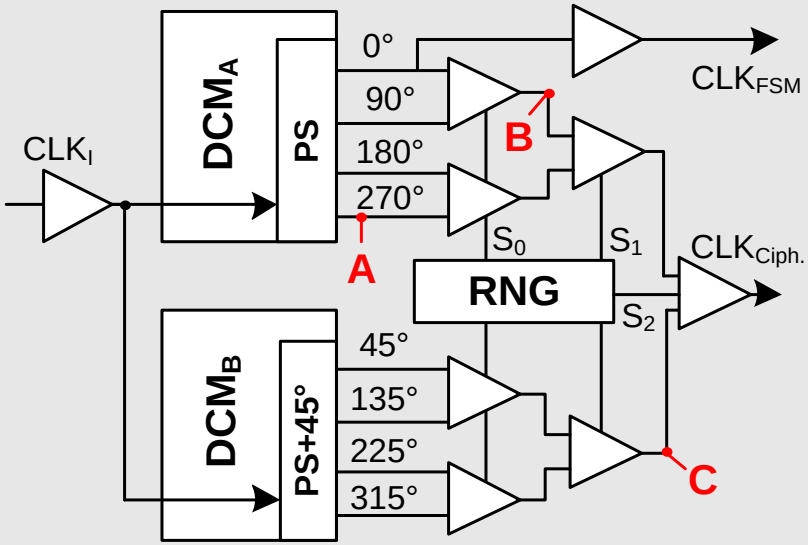
Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - **Disaligning Clocks**
  - Power Equalization
  - Memory Masking
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- Conclusions

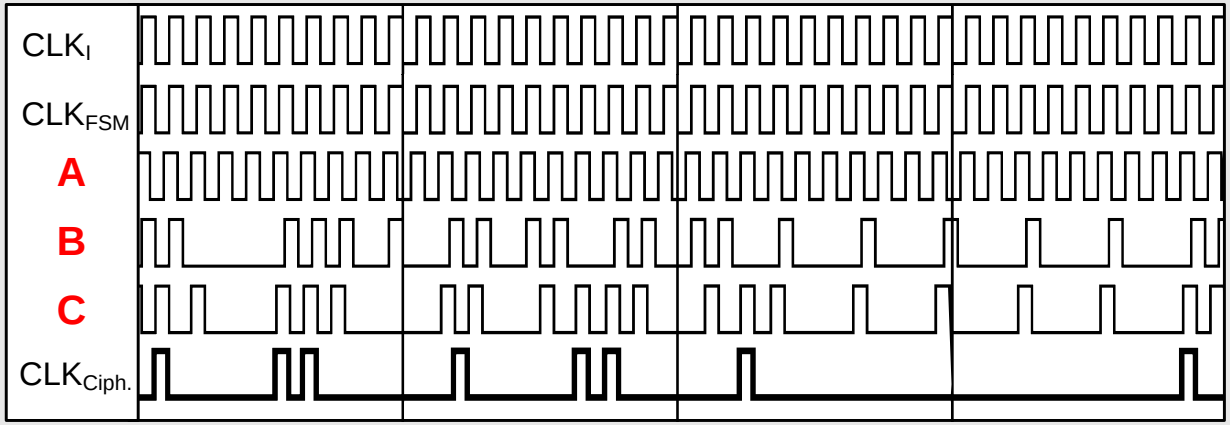


Clock Disalignment using DCMs

- Digital Clock Managers (DCM) support concurrent phase-shift channels
- Clock buffers can be configured as glitch-free clock multiplexers
- Cascading clock muxes result in a randomly delayed, phase-shifted clock

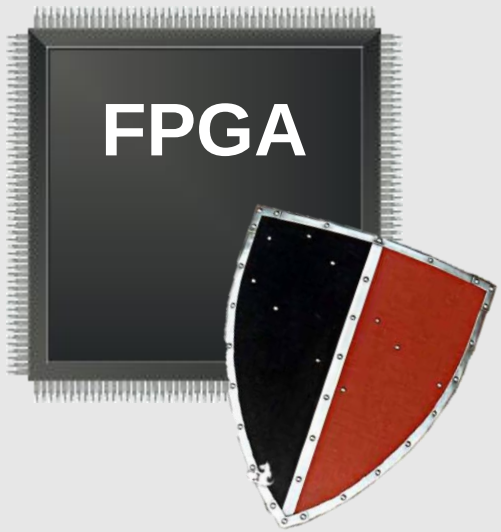


Clock Output Waveform



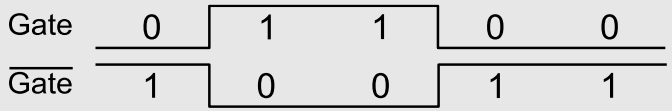
Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - Disaligning Clocks
  - **Power Equalization**
  - Memory Masking
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- Conclusions



Power Equalization by Logic Duplication

- Decorrelate power consumption from processed data by adding dual logic for power equalization



- **Problems to solve:**

Early Evaluation:

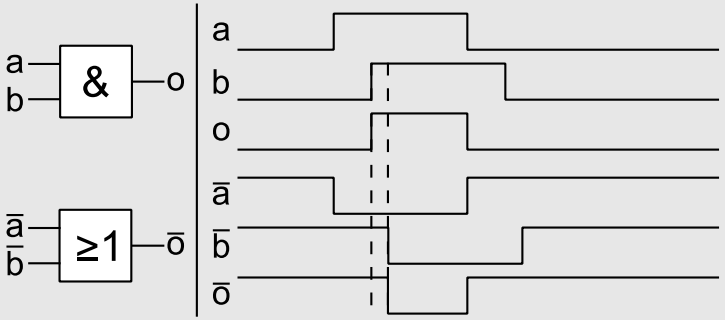
- Transition based on the arrival time of signals
- Different and data dependent points of evaluation

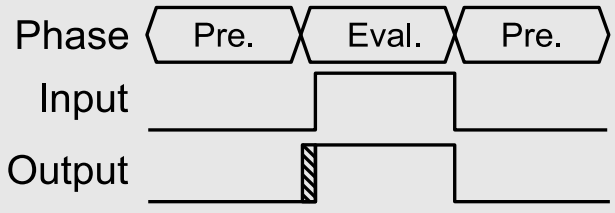
Glitches:

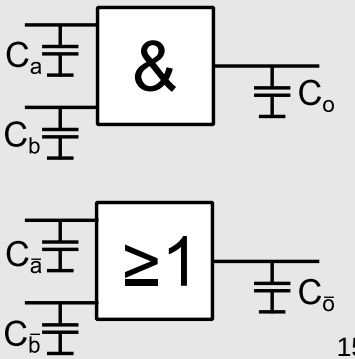
- Input signals change during evaluation.

Signal delays/capacity:

- Differences on wires/routes







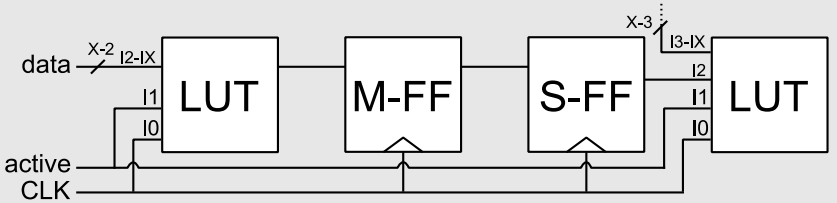
Power Equalization by Logic Duplication

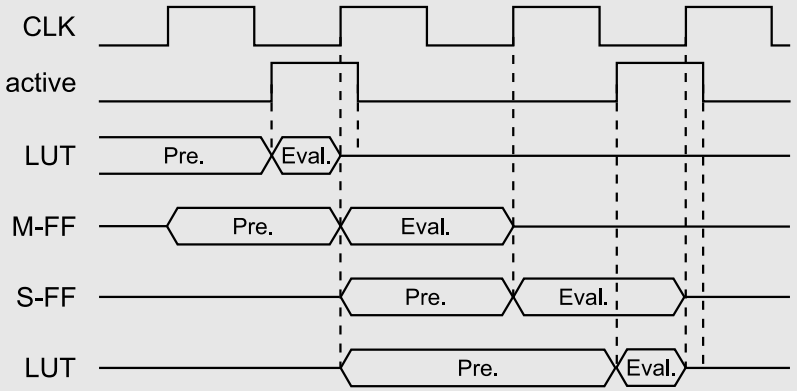
General concept:

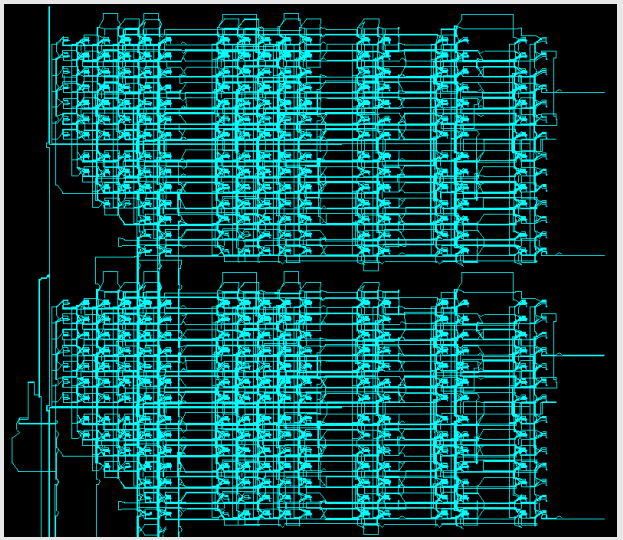
- Control signals (active, clk) connected to each LUT
- LUTs evaluated on active=HI and clk=LO
- LUTs are separated by MS-FF
- Copy placement and routing
- Dual design with complementary LUT logic

$$o = active \cdot \overline{CLK} \cdot f(I2, \ldots, IX)$$

$$\bar{o} = active \cdot \overline{CLK} \cdot \overline{f(\bar{I2}, \ldots, \overline{IX})}$$

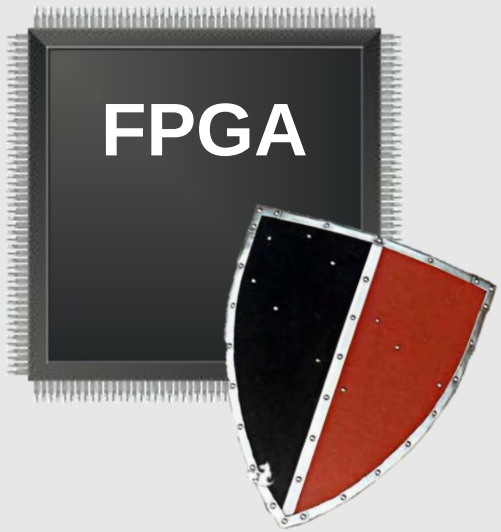






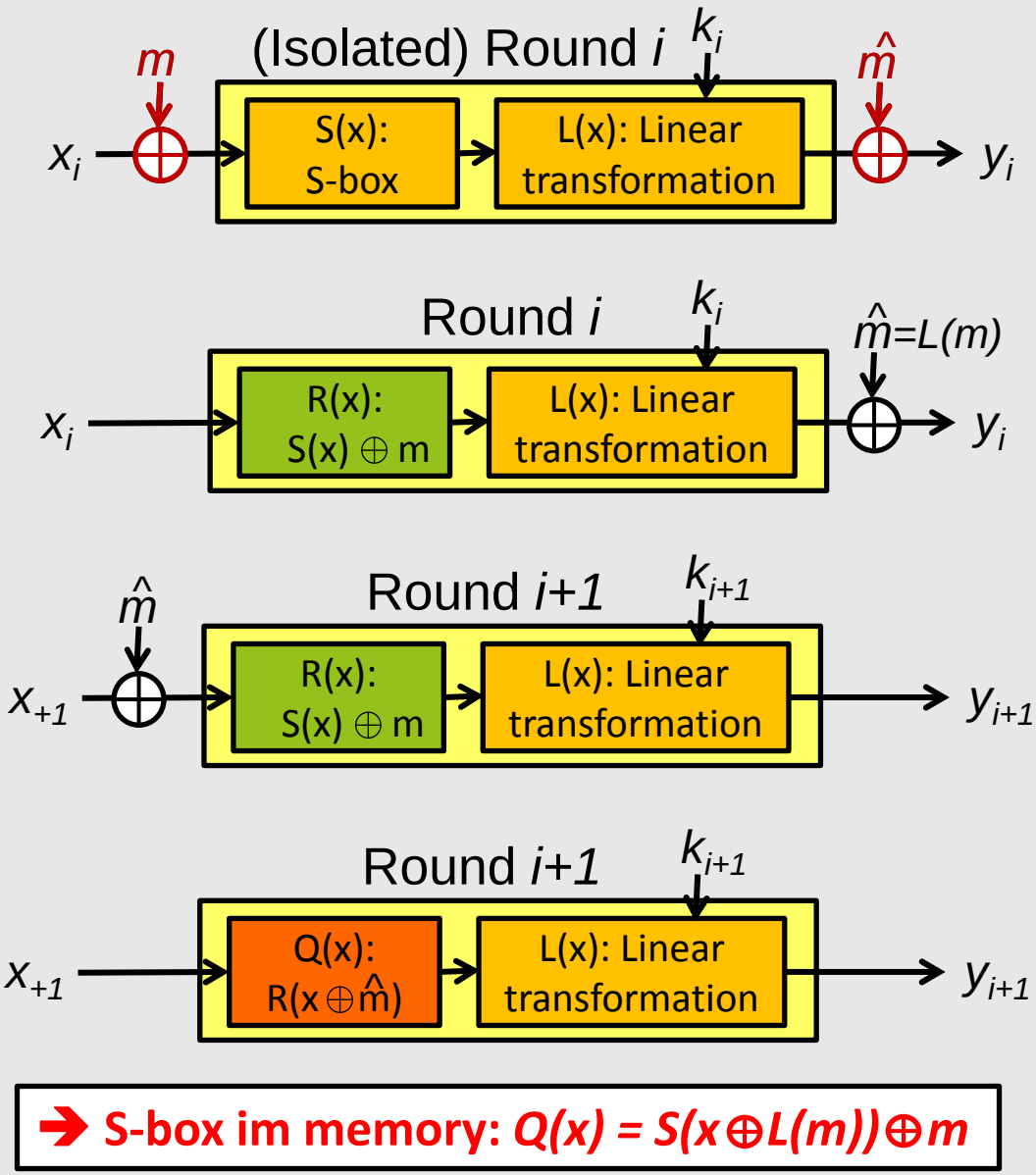
Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - Disaligning Clocks
  - Power Equalization
  - **Memory Masking**
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- Conclusions



Proposal #5: Data Masking with BRAMs

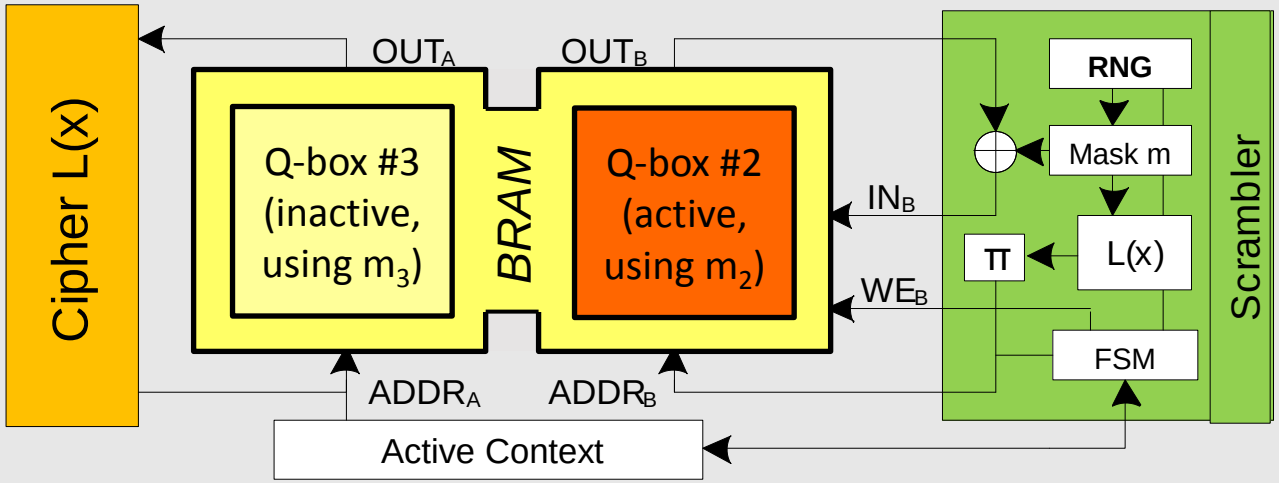
- Round functions often have linear and non-linear part (S-box in memory)
- CM: implement masking on data path
- **Implementation idea:**
  - Push masking scheme into dual-ported memory (S-box)
  - Perform mask update by concurrent process
- **Simplification**: use same random mask for (few) consecutive rounds
  ➔ (first-order SCA-resistance)



➔ **S-box im memory: *Q(x) = S(x⊕L(m))⊕m***

Proposal #5: Data Masking with BRAMs

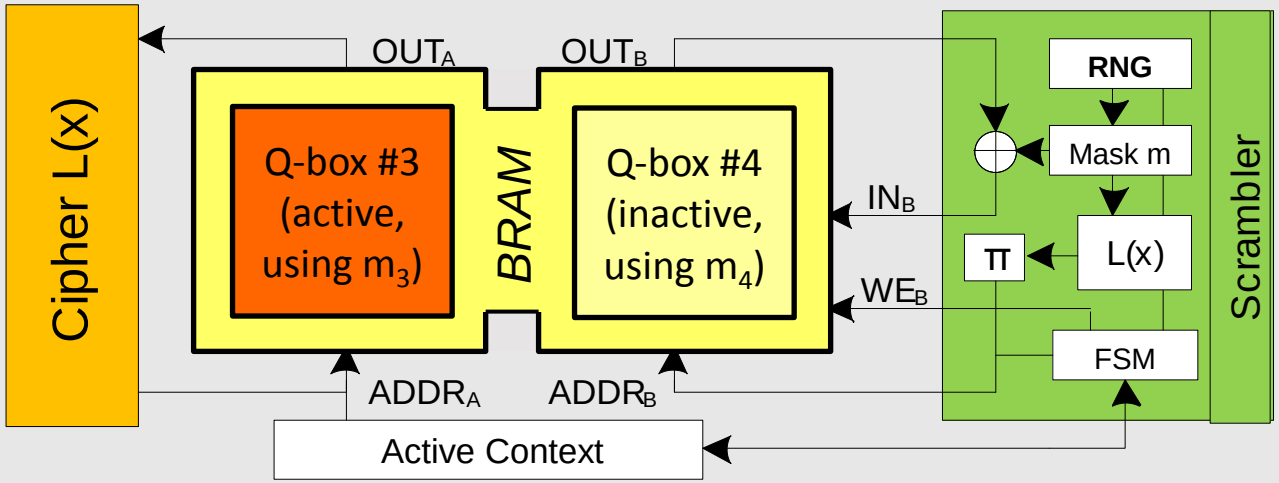
- Dual-ported BRAM allows simultaneous access and mask update in Q-box
  - Active context (Q-box #1) used by cipher operation
  - Inactive context (Q-box #2) updates mask by concurrent process
  - Context switch after update and cipher process are finished



Current Mask m:
F439AD0B8C…

Proposal #5: Data Masking with BRAMs

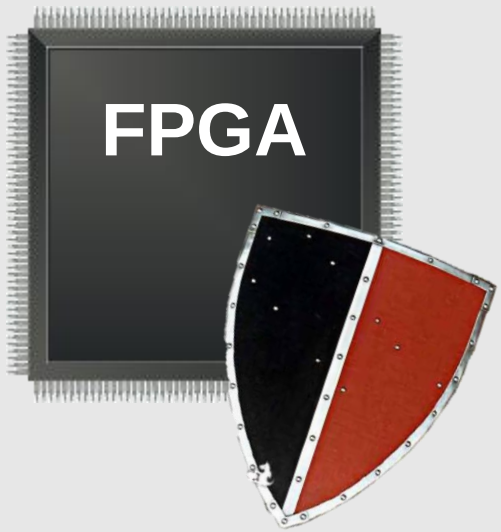
- Dual-ported BRAM allows simultaneous access and mask update in Q-box
  - Active context (Q-box #1) used by cipher operation
  - Inactive context (Q-box #2) updates mask by concurrent process
  - Context switch after update and cipher process are finished



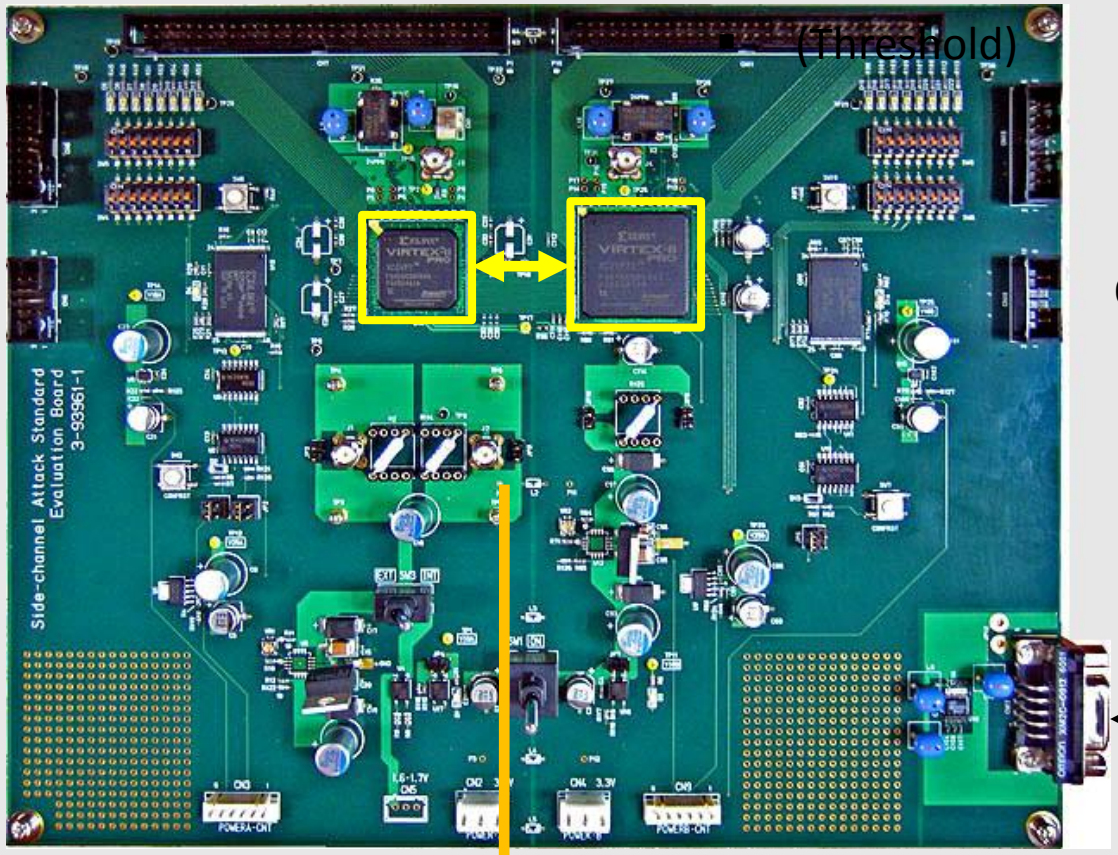
Current Mask m:
4E9A25C321…

Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - Disaligning Clocks
  - Power Equalization
  - Memory Masking
  - **Evaluation and Results**
- Physically Secure Systems on pSoCs
- Conclusions



High-Performance Side-Channel Evaluation

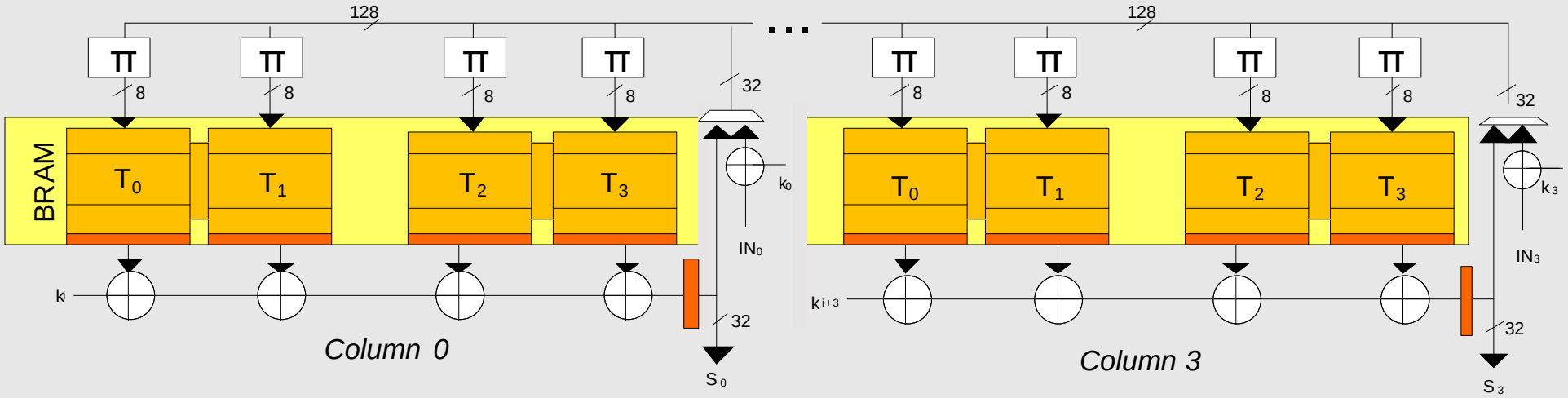


Speed of the measurement
depends on the length of each trace
In this case, 2000 points,
100M traces in 11 hours!

UART

PC sends a small number of bytes (~20)
Control FPGA communicates with the Target FPGA
sending/receiving ~10K plaintext/ciphertext
while the oscilloscope measures

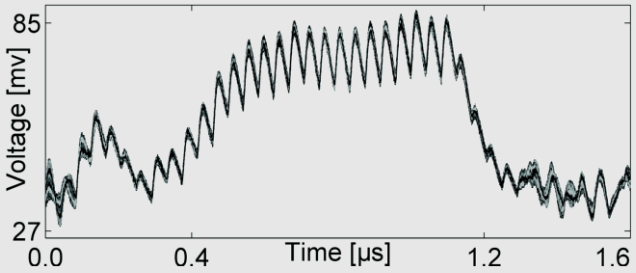
Evaluation based on AES T-Table Implementation

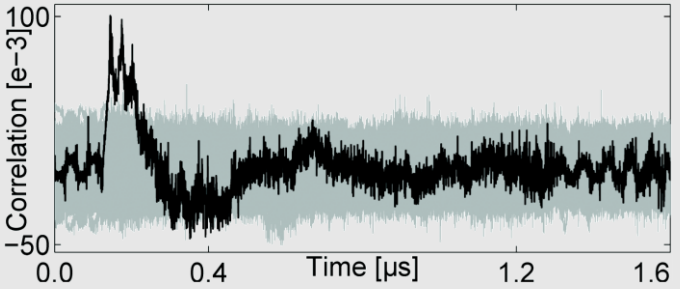


- AES-128 T-Table implementation/128-bit data path (16 T-Tables, 21 cycles)
- SASEBO board populated with Xilinx Virtex-II Pro FPGA (xc3vp7)
- Measuring setup: Diff. Probe at LeCroy WP715Zi 1.5 GHz@2GS/s
- Correlation Power Analysis (CPA) using Hamming Weight (HW) model

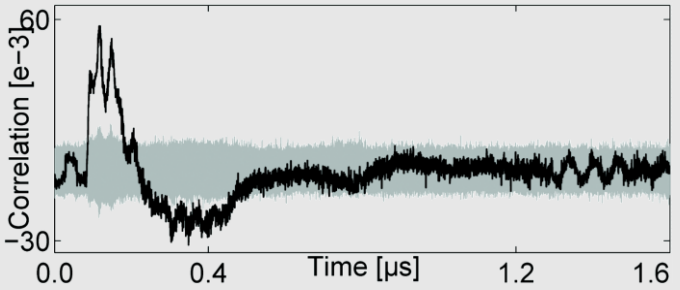
Evaluations: CPA on individual CMs

Plain AES-128@24Mh:
10^4 measurements
→ 3,000 traces req.

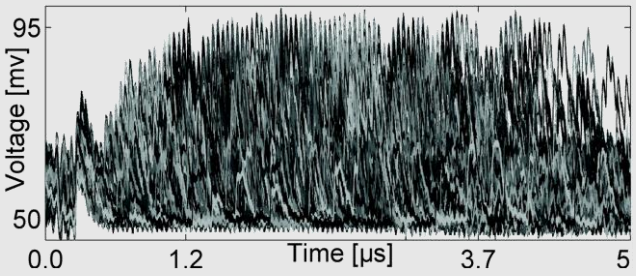


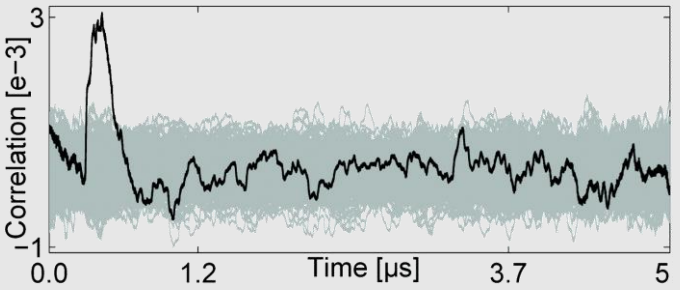


Individual/all noise generators combined:
Parameters used: *r=16 (instances), s=36 (width)*
5x10^4 measurements
→ 8,000 traces req.

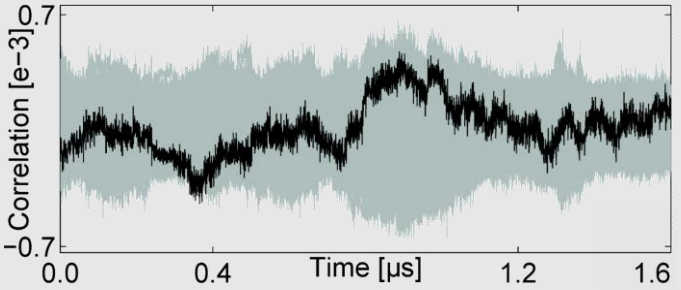


Clock disalignment
8 phase shift steps
10^7 measurements
→ 3,000,000 traces req.





Memory masking with dual-ported BRAMs
10^8 measurements
→ Not successful yet (with first-order attack);
DPA v4 contest



Evaluations: Efficiency and Resources

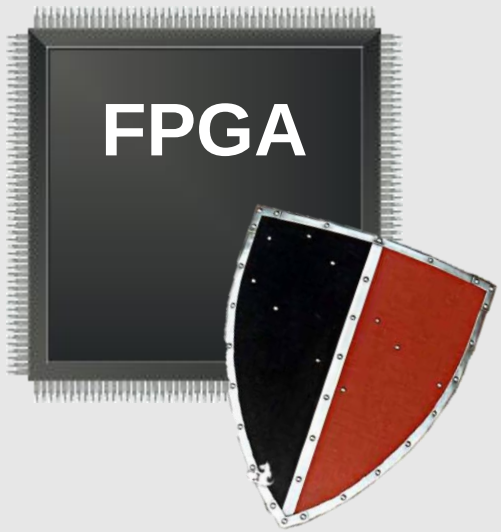
- To achieve higher SCA protection, combine several countermeasures
- CMs are quite efficient (parameters used: *s=16 (instances), r=36 (unit width))*

Proposal/ Method	**Overhead for AES T-Table Case Study** *Logic*	*Time*
# 1: SRL16	576 LUT	*none*
# 2: Write Collisions	16 BRAM, 576 LUT	*none*
# 3: Short Circuits	48 LUT	*none*
# 4: Clock Disalignment	1 DCM, 7 CB	3.77×
# 5: Clock Manip. Det.	3 LUT, 2 FF	*none*
# 6: Memory Masking	8 BRAM, 1706 LUT, 1169 FF	*none*

(FF = Flip-Flop, LUT = Look-Up-Table, CB = Clock Buffer,
DCM=Digital Clock Manager, BRAM = Block RAM)

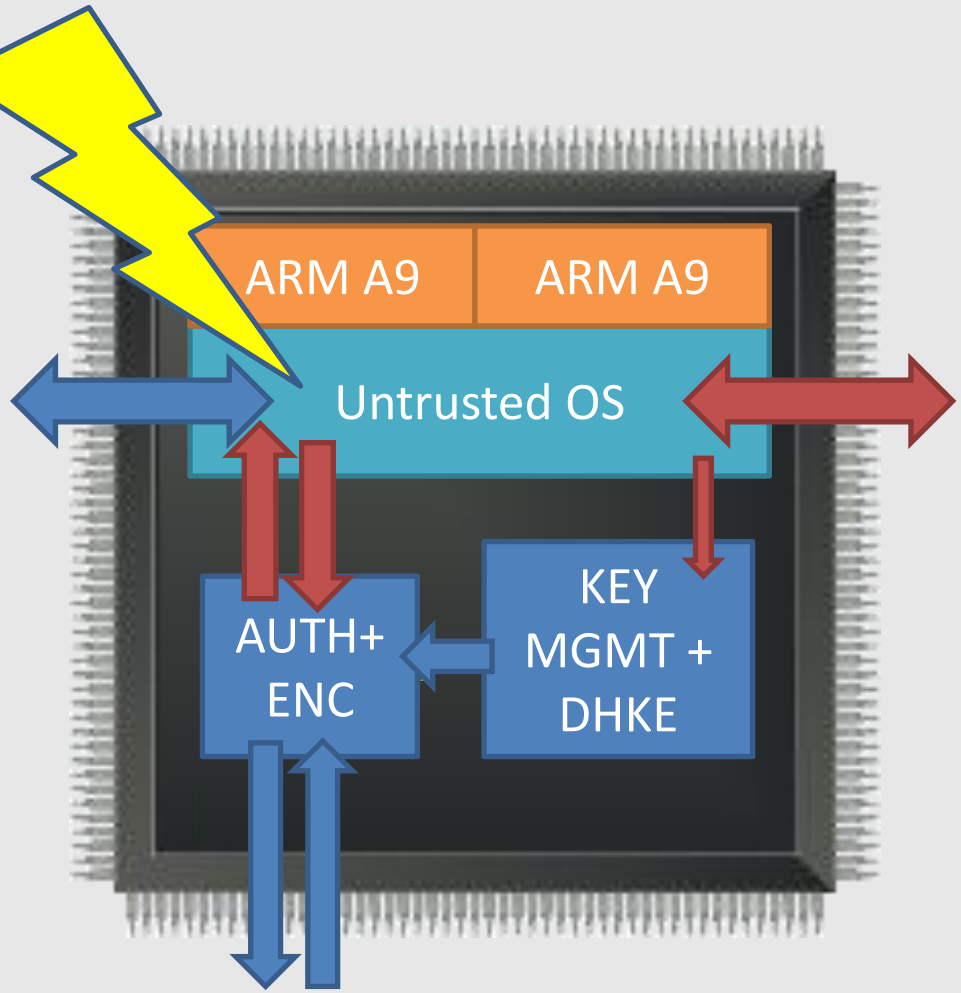
Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - Disaligning Clocks
  - Power Equalization
  - Memory Masking
  - Evaluation and Results
- **Physically Secure Systems on pSoCs**
- Conclusions



A) Secure I/O for Programmable SoCs

- **Operating systems are hard to protect against all attacks**
  - Most OS are too complex for formal security verification
- **Idea**: Move security components into a minimal hardware kernel
  - OS handles encrypted data only
  - Keys are embedded in hardware
  - if OS is compromised, attacker cannot access secret data
  - Update of security components by reconfiguration
- **Proof-of-concept:** VOIP encryption on Zedboard (Zynq)

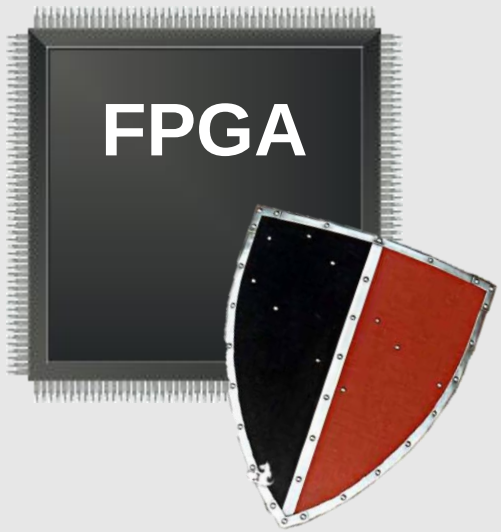


Secure VoIP Implementation on Zedboard

Proof-of-concept: VOIP encryption on Zedboard (Zynq 7020)

Agenda

- Introduction and Motivation
- **Countermeasures for Reconfigurable Hardware**
  - Noise Generation
  - Disaligning Clocks
  - Power Equalization
  - Memory Masking
  - Evaluation and Results
- Physically Secure Systems on pSoCs
- **Conclusions**



Conclusions and Outlook

- **Cryptography in reconfigurable hardware require protection against different types of physical attacks**
- **Generic countermeasures for FPGAs to prevent side-channel attacks**
  - Tailored to work with specific FPGA resources
  - Utilizing resources that would be wasted otherwise
- **Current subjects of research:**
  - (Generic) Countermeasures against fault-injection attacks
  - Hardware obfuscation techniques for reconfigurable systems
  - Understanding back-annotation in FPGAs (for power-equalization)

Thank you!